Application of the LSPI reinforcement learning technique to co-located network negotiation

Milos Rovcanin
Ghent University - iMinds, Department of Information Technology (INTEC)
Gaston Crommenlaan 8, Bus 201, 9050 Ghent, Belgium
Email: milos.rovcanin@intec.ugent.be

***Abstract*—Optimizing multiple co-located networks, each with a variable number of network functionalities that influence each other, is a complex problem that has not yet received a lot of attention in the research community. However, since independent co-located networks increasingly influence each other, optimization solutions can no longer afford to look only at the performance of a single network. To this end, we propose a multi-tiered solution, based on Least Square Policy Improvement (LSPI), a machine learning technique.**

***Index Terms*—Self-learning, network optimization, reasoning engine, LSPI;**

I. Introduction

Cognitive networks consist of decision making entities capable of planning actions according to the observed data and taking appropriate steps towards their execution [1]. Gathering feedback upon completion of all the planned tasks helps evaluate the effects of previously taken decisions and improves the decision making policy (see Fig. 1). As a result, human involvement during the operational lifetime is reduced to a minimum.

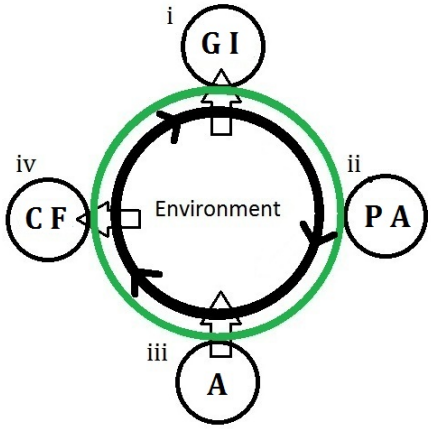

Fig. 1. General concept of a cognitive cycle: 1 - gathering necessary information; 2 - planning actions; 3 - taking actions; 4 - collecting feedback for evaluation

The same concept can be used during the process of co-located network cooperation. However, before engaging into cooperation, networks should "know" their optimal configuration setup. In other words, considering their high level goals, networks must be able to adjust properties of every protocol that is being used and decide upon what combination of available protocols (services) maximize their performance. This way, all the participants in the cooperation will be able to estimate the potential benefits and costs of the eventual cooperation.

Our research gives major guidelines of how to use the LSPI technique [5] for the purpose of creating a self-aware, self-configuring and self-optimizing network [2].

II. Methods and discussion

A. SymbioNets paradigm

The starting point of our research is the SymbioNets networking paradigm [3]. It proposes and describes different forms of network cooperation:

- Sharing of information, such as environment information or spectrum information;
- Sharing of infrastructure such as processing capacity or the sharing of each other nodes for routing purposes;
- Sharing of (networking) services, can be offered to each other, such as positioning, synchronization, address translation, QoS functions, code updates, security provisions or internet connectivity.

At the current stage of research, we emphasize on the step that has to be performed prior to initiation of a cooperation - each participating network needs to learn the effects of activating network services on its own performance, regarding its high level goals (incentives). What justifies our statement is the fact that once the network "knows" its optimal configuration, it can take a firm point during the process of negotiation with co-located communities. Reasoning entities, proposed as initiators and controllers of the network cooperation process are clearly the points in a network where the LSPI mechanism can be implemented.

B. LSPI basics

LSPI is a reinforcement based, machine learning technique [4]. LSPI reasoning agent(engine) learns by maximizing its *state-action function*, usually referred to as the Q-function [6]. Its formal outlook is known as the Bellman's equation:

$$Q(s,a) = r(s,a) + \gamma \sum_{s'} P(s'|s,a) max Q(s',a') \quad (1)$$

An agent updates Q-values of each state/action pair once it switches from state s to s', upon utilizing action a. At each state it uses the same criteria to choose the best possible action

- it picks up the one which has the highest Q-value. That way, certain decisions will be enforced.

Within the LSPI, Q-values are approximated with a linear parametric combination of k *basis functions*, also referred to as *features*:

$$Q(s,a;w) = \sum_k \phi_j(s,a)\omega_j \quad (2)$$

Argument ω_j is the weight parameter. Basis functions are arbitrary and generally non-linear functions of s and a. It is important to make them linearly independent to ensure that there are no redundant parameters. Generally, the number of basis functions is significantly smaller than the number of state/action pairs.

Combination of equations (1) and (2) results in a linear system:

$$\begin{gathered} \omega = A^{-1}b \\ \text{where: } A = \Phi^T(\Phi - \gamma P^\pi \Phi) \\ b = \Phi^T R \end{gathered} \quad (3)$$

Both A and b matrices are populated by collecting samples (s, a, s', r) from the environment. The parameters s, a, s' and r are the current state, action, new state and immediate reward, respectively. Using L number of samples, we can construct an approximate version of $\hat{\Phi}$, $\hat{P^\pi}$ and $\hat{R}$ and recalculate the ω_j factors for each basis function.

Main reasons for using the LSPI rather than any other machine learning approach are:

- It converges faster than all other known algorithms, since the samples are used more efficiently.
- It does not require fine tuning of the initial parameters such as *learning rate*.
- LSPI learns the weights of the linear functions and updates Q-values based on the most updated information regarding the features, while in other approaches agents make decisions directly based on Q-values, which may be outdated, depending on the network dynamics.

III. Results and challenges

We have managed to design and implement an LSPI based system that determines the optimal set of services for a specific network, in regards to its high level goals. Given any possible set of services that a network can provide and any number of high level goals, the LSPI reasoning engine will calculate the optimal setup.

LSPI's main advantages to other forms of reinforcement learning are:

- How to avoid large overhead when collecting statistics (basic functions) from the network
- What is the optimal period between changing from one state to another (learning episode)

There is no general answer to neither of those questions. Both issues are case specific and the best solution to any of them will depend on both the capabilities of the network and the metrics that are needed to calculate values of the basis functions and rewards.

In regards to determining the optimal duration of the learning period: two factors should be taken into account:

- The dynamics of the network
- The total length of a learning process

The number of episodes and their duration will directly define the duration of the entire learning process. In cases where it is possible to perform an "off-line" learning (on test beds or using network simulators), using artificially generated traffic or by recreating operational conditions, time will not be a factor. However, we must assume that, generally, this will not be possible. In those cases, determining the duration of the learning process will pose a great challenge to a system designer, since the practical value of the reasoning method will be directly evaluated through it.

A. Future development

The application of LSPI can be expanded in two directions, eventually leading to a tiered paradigm of network optimization:

1) LSPI can be used to first optimize the parameters of each network protocol of the network.
2) Next, LSPI is used to identify a number of service sets that offer acceptable network performance.
3) Finally, LSPI selects amongst the acceptable service sets those that are also beneficial for co-located networks.

It is reasonable to expect that further implementation will add new challenges to the ones that have already been identified.

IV. Conclusion

We strongly believe that the problem of interfering co-located networks will only increase. As such, innovative cross-layer and cross-network solutions that take these interactions into account, like the one proposed in this paper, will be of great importance to the successful development of efficient next-generation networks in heterogeneous environments.

References

[1] R. W. Thomas, L. A. DaSilva and A. B. MacKenzie, "Cognitive networks", Proc. IEEE DySPAN 2005, pp.352-60

[2] Wakamiya, N.; Arakawa, S.; Murata, M.,"Self-Organization Based Network Architecture for New Generation Networks", 2009 First International Conference on Emerging Network Intelligence, pp.61-68, 11-16 Oct. 2009

[3] E. De Poorter, B. Latre, I. Moerman and P. Demeester, "Symbiotic networks: Towards a new level of cooperation between wireless networks", Published in Special Issue of the Wireless Personal Communications Journal, Springer Netherlands, 45(4):479-495, June 2008

[4] T. G. Dietterich, and O. Langley, (2007) "Machine Learning for Cognitive Networks:Technology Assessment and Research Challenges in Cognitive Networks: Towards Self Aware Networks", John Wiley and Sons, Ltd, Chichester,UK. doi: 10.1002/9780470515143.ch5

[5] M. Lagoudakis and R. Parr. "Model-free least-squares policy iteration". In Proc. of NIPS, 2001.

[6] L. P. Kaelblign, M. L. Littman, A. W.Moore,"Reinforcement learning: A Survey", Journal of Artificial Intelligence Research 4 (1996) 237-285